Experimental Evaluation of Control Policies for Segway Robot in Dynamic Environments

Ian Buckley, Niharika Arora, Varun Murali

***Abstract*—**

I. INTRODUCTION

Navigation is a fundamental concern for mobile robots. The ability to successfully avoid obstacles, both static and dynamic, while moving towards a goal location largely determines the utility of a mobile robotic platform. It is important that robots avoid obstacles for their own safety as well as the safety of the people around them. Given the importance of navigation, it is not surprising that the problem has long been considered and explored; furthermore, it is reasonable to assert that approaches to the problem are as varied as the disciplines related to robotics themselves.

In this paper, control Lyapunov and barrier functions are used for robot navigation and collision avoidance. We define constraints important for navigation like distance from obstacle, minimum and maximum velocity, minimum and maximum rotational velocity, and velocity as a function of distance from obstacle for safety to generate a navigation law for the robot.

Nonlinear control based navigation is also a studied problem such as by Ting et al. [1]. Generating a policy in a reactive fashion reduces the complexity of the computation needed. Traditional path planners can be used to provide a high level trajectory by having some previous metric knowledge of the space. Lower level reactive controllers can be used for local refinement of the global path and can be used to control the robot. This reduces the complexity of re-planning every time a change in the environment is observed. Potential field based algorithms have been used to generate such reactive policies in the past with some success. There are some known pitfalls for this method with regards to local minima but several solutions for this have been proposed.

Given the possibility of having a global metric representation of a domestic environment, we generate a global trajectory and local non-linear controller that obeys the bounds imposed on the robot. If the barrier function comprises of the above information, we can make claims guarantees about the stability of the robot with respect to those objectives.

The navigation law will first be tested and verified in simulation. MATLAB will be used for this purpose. Once the simulations verify the algorithm, the controller will be implemented in the ROS framework and evaluated in a robot simulation environment (Gazebo). The final experiments will be performed on the mobile robot Jeeves, shown in Figure 1, which will be used as a test platform for the real world environment. We plan to study the law developed in both static and dynamic environments.

Fig. 1. Jeeves - A modified Segway based robot mounted with a Primesense, Microsfot Kinect, Roboteye RE05, Hokuyo LTM-30x laser and the UR5. Currently modified to include the casters for static stability

Fig. 2. Proposed modification to the Segway base inspired by the Segway robot at the University of Michigan

II. SYSTEM OVERVIEW

The robot platform shown in Figure 1 consists of a Segway RMP-200 mobile base, which has been modified to hold various sensors. The Seqway RMP 200 has been modified to be statically stable but for the purposes of this project, the casters shall be removed and 8020 adjustments will be made on the side of the robot to prevent it from tipping (inspired by Figure 2). The sensor suite on the robot consists of a Roboteye RE05 sensor capable of scanning in 3D, a primesense RGB-D camera and a Kinect RGB-D. The Hokuyo LTM-30x is mounted at the base and used for base obstacle avoidance.

In the scope of this project, the Hokuyo LTM-30x is used to detect obstacles from a planar scan. The closest laser hit can be considered an obstacle and dynamic obstacles can be modelled in this way. A microstrain IMU is mounted on the robot to estimate the state of the robot.

The robot is controlled using a PS3 controller when in tele-operation mode, which has a manual override and a dead-man switch to kill the robot due to any type of failure. The robot is also permanently connected to a computer that is monitoring the robot and can be used to kill running programs on the robot.

The dynamics (discussed in [2]) of the Segway RMP base can be modelled as a cart with an inverted pendulum which can be modelled as:

$$mLsin(\theta)\dot{\theta}^2 + (M+m)\dot{v} - F - mLcos(\theta)\ddot{\theta} = 0 \quad (1)$$

where θ is the angle of the pendulum with respect to its vertical position, v is the speed of the base, F is the force on the cart, M is the mass of the cart and m is the mass of the pendulum and L is the length of the massless rod.

The control law will further be used to test how it affects the dynamics of the system. We will measure the translation and the rotational velocities and the tip angle to verify that the control policy doesn't force the robot to go beyond its boundary conditions and ensures its safety.

III. Methods

To design the fundamental control laws explored by this paper, the robot was assumed to be modelled by unicycle dynamics, presented in the familiar rectangular form:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} -v\ \cos(\theta) \\ v\ \sin(\theta) \\ \omega \end{pmatrix},$$

where (x, y) are the global coordinates of the robot, and θ is the angle formed between the heading of the robot and the positive x-axis–v and w are the linear and translational velocities respectively. While control of the unicycle robot is possible using rectangular dynamics, a transformation of the dynamics into polar form simplified the dynamics for formulating the control law, where $r = \sqrt{(x^2+y^2)}$ is the distance between the goal and the robot, and θ is unchanged. The polar coordinates are transformed yet again into an egocentric coordinate system centered on the robot to represent the heading of the robot δ; this transformation was inspired by [3]. The final form of the dynamics are given by:

$$\begin{pmatrix} \dot{r} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} -v\ \cos(\delta) \\ \frac{v}{r}\ \sin(\delta) \end{pmatrix}. \quad (2)$$

$$\dot{\delta} = \frac{v}{r}\ \sin(\delta) + \omega \quad (3)$$

The dynamics of the robot can be represented in affine form ($\dot{x} = f(x) + g(x)u$), where

$$u = \begin{pmatrix} v \\ \omega \end{pmatrix},$$

and it is obvious that the drift term is zero. The following intermediate developments are similar to those in [3], but are presented here for completeness and to highlight the unique approach taken by this paper. To design the control law, the quintessential quadratic Lyapunov function candidate is considered:

$$\begin{aligned} V &= \frac{1}{2}(r^2 + \theta^2) \\ \dot{V} &= r\dot{r} + \theta\dot{\theta} \\ &= -rv\ \cos(\delta) + \theta v\ \sin(\delta) \end{aligned}$$

Fixing δ to enforce a virtual steering control as in [3],

$$\delta = \arctan(\theta).$$

The Lyapunov candidate becomes:

$$\dot{V} = -rv\ \cos(\arctan(\theta)) + \theta v\ \sin(\arctan(\theta)). \quad (4)$$

Because $\theta \in (-\pi, \pi]$, $v > 0$, and $r \geq 0$, $\dot{V} < 0\ \forall\theta, r \neq 0$; thus, the steering control asymptotically drives $r, \theta \to 0$. Furthermore, by the definition of the steering control, $\delta \to 0$ as $\theta \to 0$, so the virtual steering control not only drives the robot to the final position, but aligns the heading of the robot with the positive x-axis by the time it arrives at the final position.

IV. Results

A. Simulation

References

[1] L. Ting, C. J. Macnab, and S. Magierowski, "Reactive robot navigation utilizing nonlinear control," *International Journal of Advanced Robotic Systems*, vol. 11, 2014.

[2] A. Castro, "Modeling and dynamic analysis of a two-wheeled inverted-pendulum," 2012.

[3] J. J. Park and B. Kuipers, "A smooth control law for graceful motion of differential wheeled mobile robots in 2d environment," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 4896–4902.